SUBJECT: COMPUTER SCIENCE							
DCS	BCS-C101	Object- Oriented Programming in C++	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Students are expected to have some basic knowledge about computers, some knowledge in programming language is preferred.

Course Objectives:

- To design, analyze and evaluate computer programs using the C++ programming language.
- To apply object-oriented programming principles and techniques using C++.

Course Outcomes:

CO1	Describe the procedural and object-oriented paradigm with concepts of streams, classes, functions, data and objects.
CO2	Understand dynamic memory management techniques using pointers, constructors, destructors, etc
CO3	Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
CO4	Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
CO5	Demonstrate the use of various OOPs concepts with the help of programs.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to C and C++** History of C and C++, Overview of Procedural Programming and Object-Orientation Programming, using main () function, Compiling and Executing Simple Programs in C++.	2
2.	**Data Types, Variables, Constants, Operators and Basic I/O** Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar etc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).	3
3.	**Expressions, Conditional Statements and Iterative Statements** Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)	4
4.	**Functions and Arrays** Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, accessing individual elements in an	6

	Array, manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two-Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays	
5.	**Derived Data Types (Structures and Unions)** Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.	3
6.	**Pointers and References in C++** Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values	6
7.	**Using Classes in C++** Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables & Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.	8
8.	**Overview of Function Overloading and Operator Overloading** Need of Overloading functions and operators, Overloading functions by number and type of arguments, looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)	5
9.	**Inheritance, Polymorphism and Exception Handling** Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.	8
10.	**File I/O, Preprocessor Directives** Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros	3
	Total Lectures	48

Suggested Text Book(s)

1.	H. Schildt C++, "The Complete Reference Book", McGraw Hill.
2.	E. Balaguruswamy, "Object Oriented Programming with C++", McGraw Hill.
3.	J. R. Hubbard, "Programming with C++", Schaum's Outlines, McGraw Hill.
4.	R. Albert and T. Breedlove, C++, "An Active Learning Approach", Jones and Bartlett India Ltd.

Suggested Reference Book(s)

1.	Stroustrap B., “The C++ Programming Language”, Addison Wesley.
2.	Bruce Eckel, “Thinking in C++”, Pearson.

Other Useful Resource(s)

1.	https://onlinecourses.nptel.ac.in/noc16_cs17/preview
2.	https://onlinecourses.nptel.ac.in/noc17_cs25/announcements
3.	https://www.tutorialspoint.com/cplusplus/
4.	http://www.cplusplus.com/doc/tutorial

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	2	1	-	3	3	**2.1**
CO2	3	3	2	3	3	2	2	3	**2.6**
CO3	3	3	2	1	1	-	3	3	**2.0**
CO4	3	3	2	3	3	2	2	3	**2.6**
CO5	3	3	3	2	1	1	3	3	**2.4**
AVG.	**3.0**	**3.0**	**2.2**	**2.2**	**1.8**	**1.0**	**2.6**	**3.0**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	2	3	**2.7**
CO2	3	2	3	**2.7**
CO3	3	2	3	**2.7**
CO4	3	2	3	**2.7**
CO5	3	2	3	**2.7**
AVG.	**3.0**	**2.0**	**3.0**	**2.7**